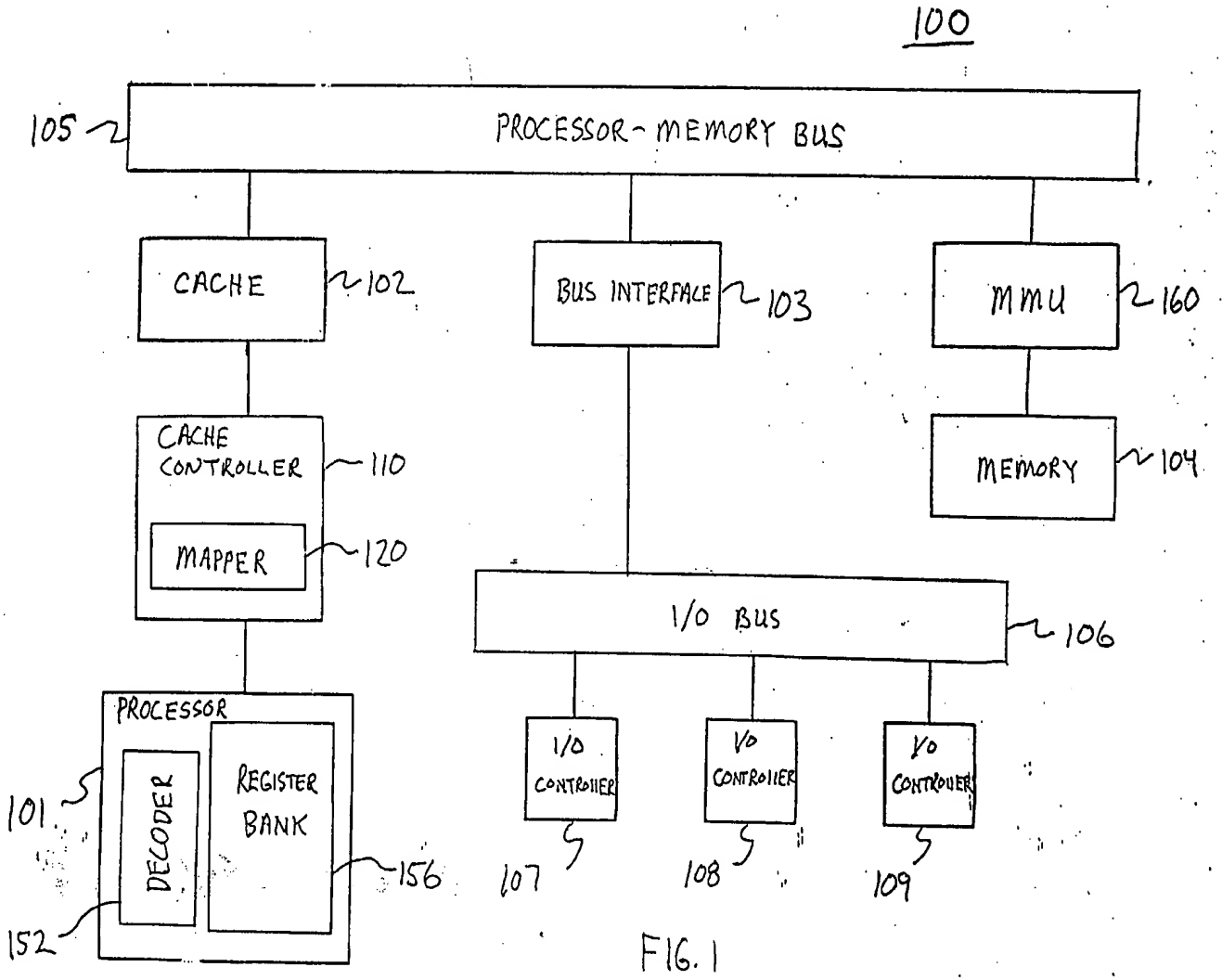

FIG. 1

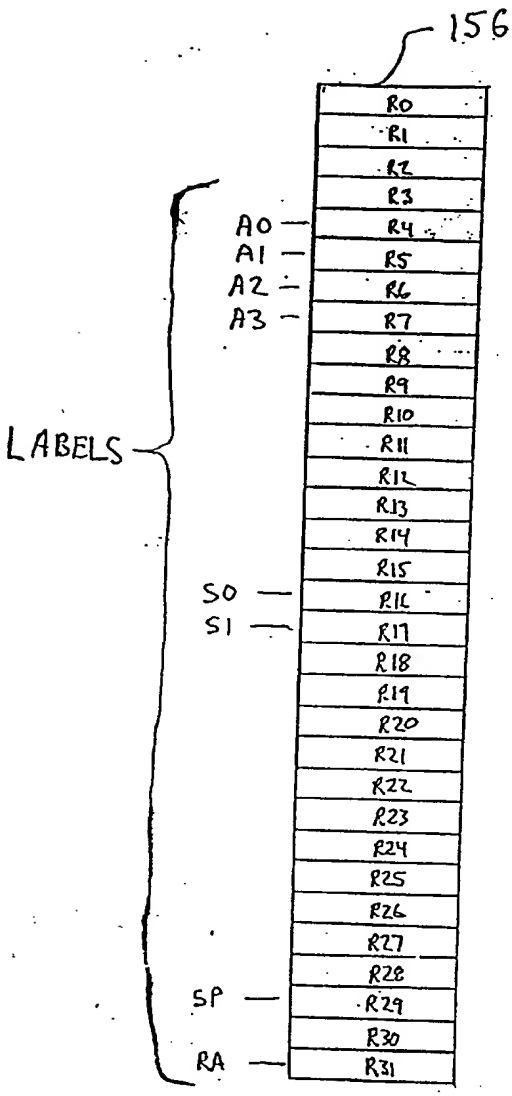
156
R0
R1
R2
R3
A0
R4
A1
R5
A2
R6
A3
R7
R8
R9
R10
R11
R12
R13
R14
R15
S0
R16
S1
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
SP
R29
R30
RA
R31
LABELS

FIG. 2

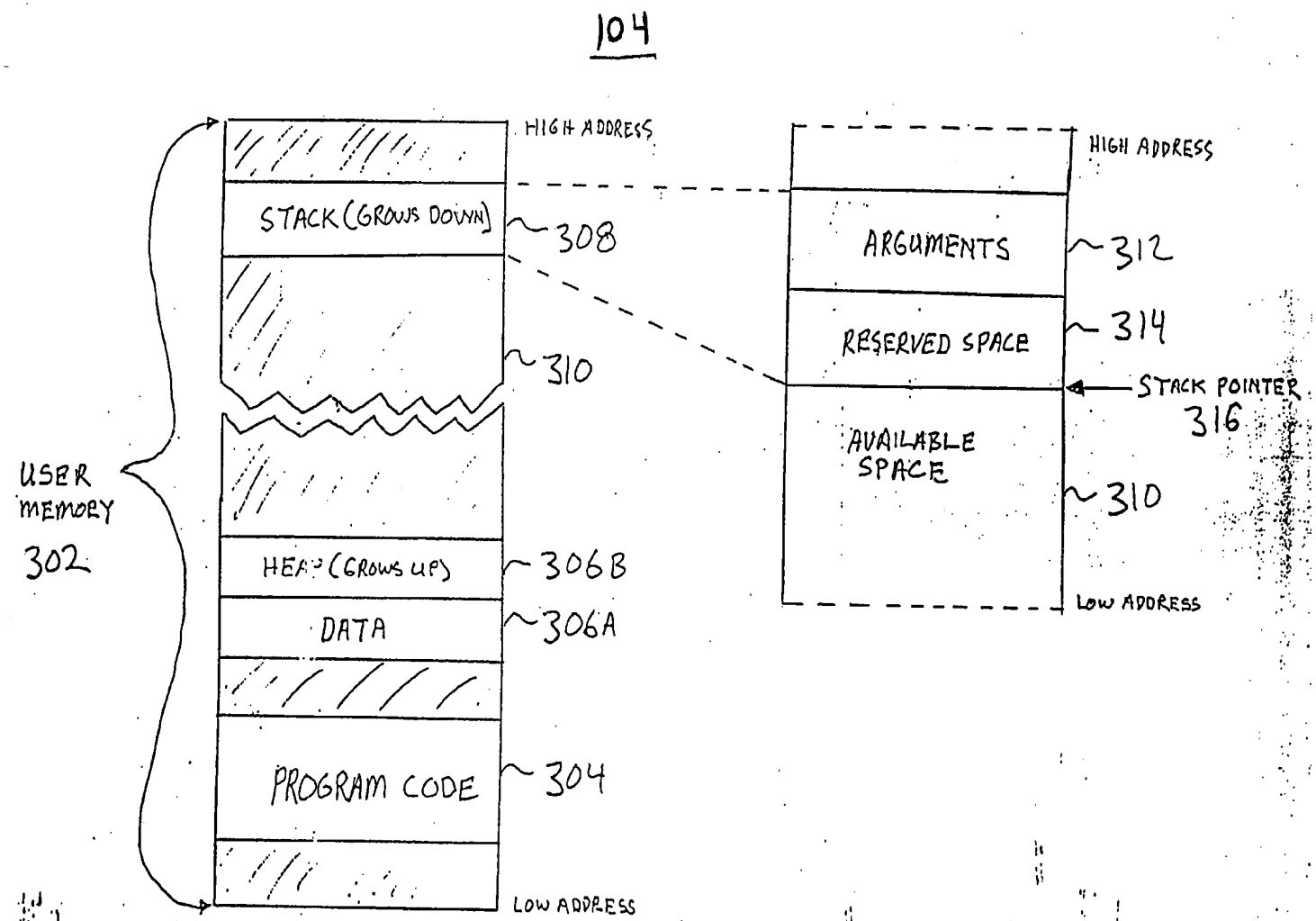

FIG. 3

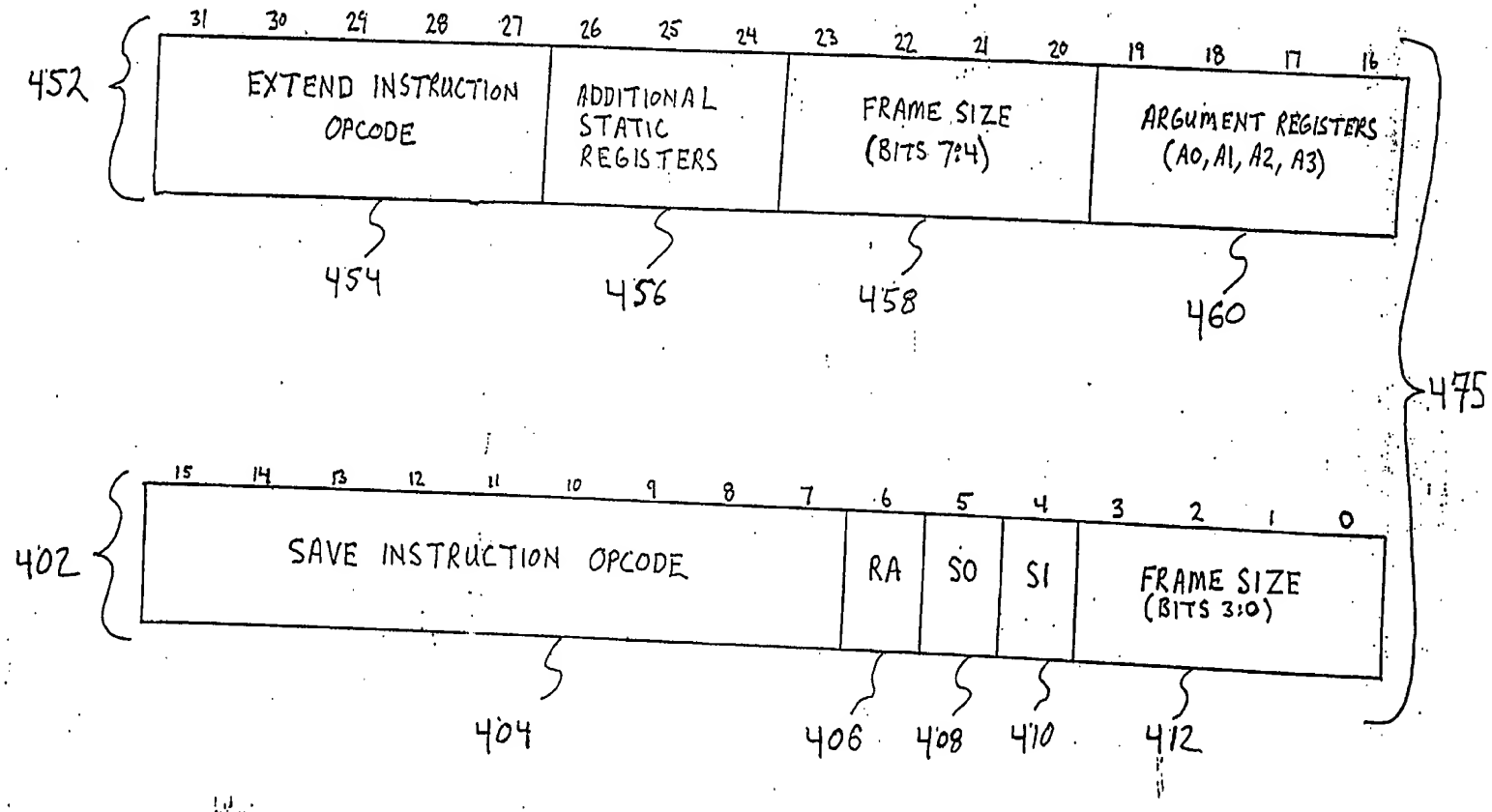

FIG. 4

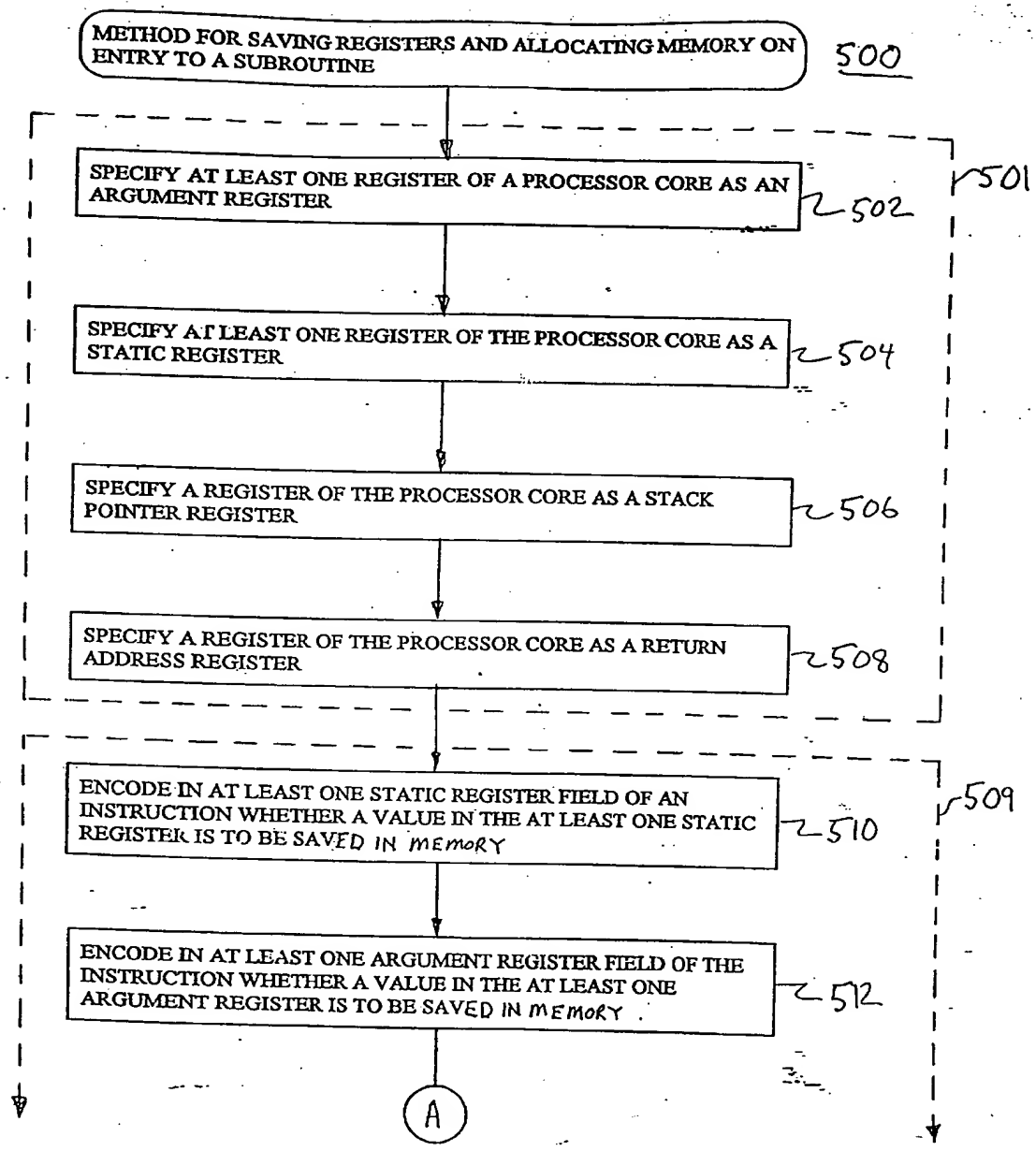
METHOD FOR SAVING REGISTERS AND ALLOCATING MEMORY ON ENTRY TO A SUBROUTINE
500
SPECIFY AT LEAST ONE REGISTER OF A PROCESSOR CORE AS AN ARGUMENT REGISTER
502
501
SPECIFY AT LEAST ONE REGISTER OF THE PROCESSOR CORE AS A STATIC REGISTER
504
SPECIFY A REGISTER OF THE PROCESSOR CORE AS A STACK POINTER REGISTER
506
SPECIFY A REGISTER OF THE PROCESSOR CORE AS A RETURN ADDRESS REGISTER
508
ENCODE IN AT LEAST ONE STATIC REGISTER FIELD OF AN INSTRUCTION WHETHER A VALUE IN THE AT LEAST ONE STATIC REGISTER IS TO BE SAVED IN MEMORY
510
509
ENCODE IN AT LEAST ONE ARGUMENT REGISTER FIELD OF THE INSTRUCTION WHETHER A VALUE IN THE AT LEAST ONE ARGUMENT REGISTER IS TO BE SAVED IN MEMORY .
512
A

FIG. 5A

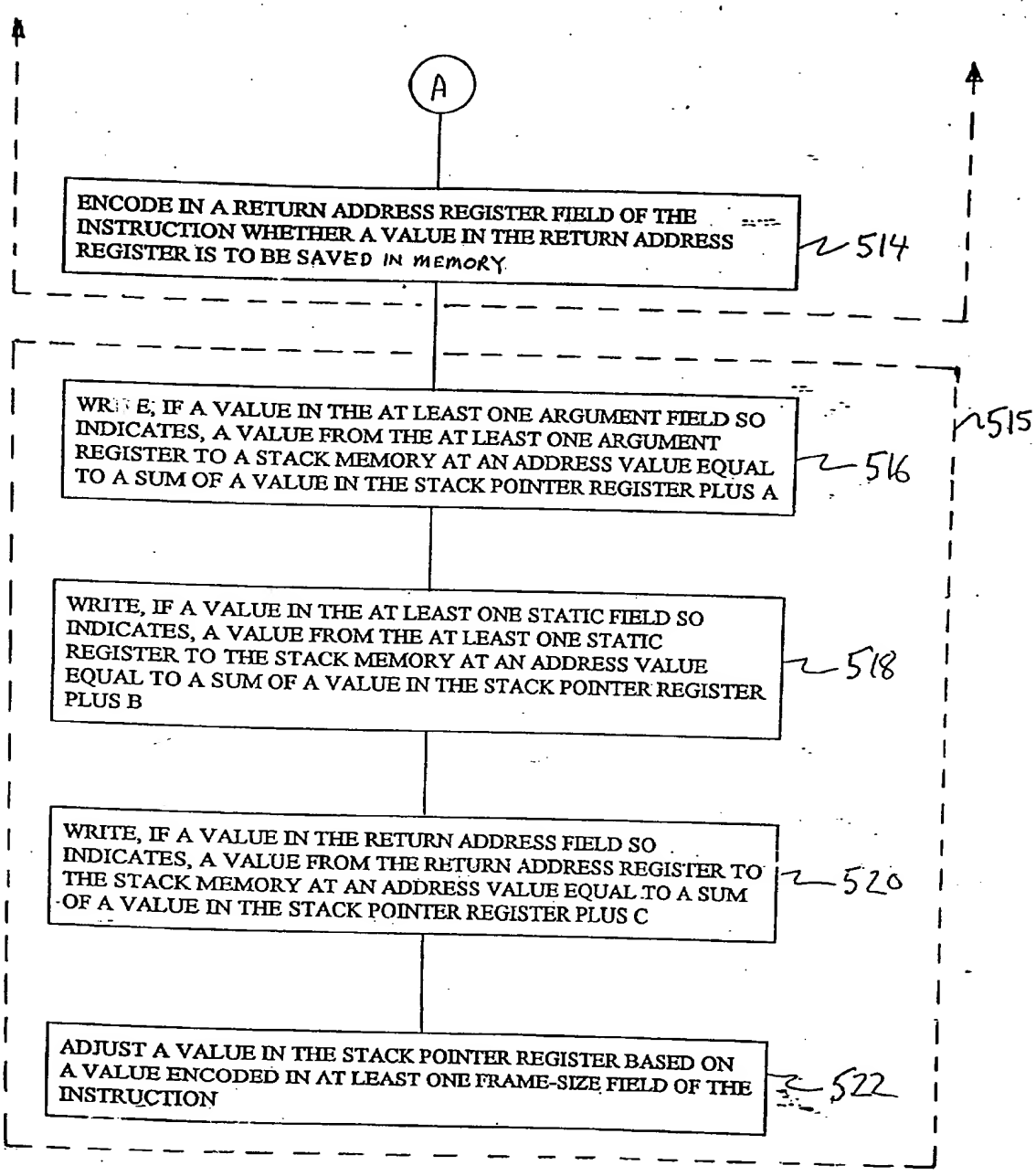
A
ENCODE IN A RETURN ADDRESS REGISTER FIELD OF THE INSTRUCTION WHETHER A VALUE IN THE RETURN ADDRESS REGISTER IS TO BE SAVED IN MEMORY
514
WRITE, IF A VALUE IN THE AT LEAST ONE ARGUMENT FIELD SO INDICATES, A VALUE FROM THE AT LEAST ONE ARGUMENT REGISTER TO A STACK MEMORY AT AN ADDRESS VALUE EQUAL TO A SUM OF A VALUE IN THE STACK POINTER REGISTER PLUS A
516
515
WRITE, IF A VALUE IN THE AT LEAST ONE STATIC FIELD SO INDICATES, A VALUE FROM THE AT LEAST ONE STATIC REGISTER TO THE STACK MEMORY AT AN ADDRESS VALUE EQUAL TO A SUM OF A VALUE IN THE STACK POINTER REGISTER PLUS B
518
WRITE, IF A VALUE IN THE RETURN ADDRESS FIELD SO INDICATES, A VALUE FROM THE RETURN ADDRESS REGISTER TO THE STACK MEMORY AT AN ADDRESS VALUE EQUAL TO A SUM OF A VALUE IN THE STACK POINTER REGISTER PLUS C
520
ADJUST A VALUE IN THE STACK POINTER REGISTER BASED ON A VALUE ENCODED IN AT LEAST ONE FRAME-SIZE FIELD OF THE INSTRUCTION
522

FIG. 5B

650

SAVE INSTRUCTION

```
temp ← GPR[29]
if ra = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[31]
endif
if sl = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[17]
endif
if s0 = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[16]
endif
if framesize = 0 then
        temp ← GPR[29] - 128
else
        temp ← GPR[29] - (0 || (framesize << 3))
endif
GPR[29] ← temp
```

FIG. 6

aregs Encoding (binary)	Registers Saved as Arguments	Registers Restored as StaticRegisters
0000	None	None
0001	None	GPR[7]
0010	None	GPR[6], GPR[7]
0011	None	GPR[5], GPR[6], GPR[7]
1011	None	GPR[4], GPR[5], GPR[6], GPR[7]
0100	a0	None
0101	a0	GPR[7]
0110	a0	GPR[6], GPR[7]
0111	a0	GPR[5], GPR[6], GPR[7]
1000	a0, a1	None
1001	a0, a1	GPR[7]
1010	a0, a1	GPR[6], GPR[7]
1100	a0, a1, a2	None
1101	a0, a1, a2	GPR[7]
1110	a0, a1, a2, a3	None
1111	Reserved	Reserved

FIG. 7

800

EXTENDED SAVE INSTRUCTION

```
temp ← GPR[29]
temp2 ← GPR[29]
case aregs of
        2#0000 2#0001 2#0010 2#0011 2#1011: args ← 0
        2#0100 2#0101 2#0110 2#0111: args ← 1
        2#1000 2#1001 2#1010: args ← 2
        2#1100 2#1101: args ← 3
        2#1110: args ← 4
        otherwise: UNPREDICTABLE
endcase
if args > 0 then
        VirtualMemory [temp] ← GPR[4]
        if args > 1 then
                VirtualMemory [temp + 4] ← GPR[5]
                if args > 2 then
                        VirtualMemory [temp + 8] ← GPR[6]
                        if args > 3 then
                                VirtualMemory [temp + 12] ← GPR[7]
                        endif
                endif
        endif
endif
if ra = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[31]
endif
if xsregs > 0 then
        if xsregs > 1 then
                if xsregs > 2 then
                        if xsregs > 3 then
                                if xsregs > 4 then
                                        if xsregs > 5 then
                                                if xsregs > 6 then
                                                        temp ← temp - 4
                                                        VirtualMemory[temp] ← GPR[30]
                                                endif
                                                temp ← temp - 4
                                                VirtualMemory[temp] ← GPR[23]
                                        endif
                                        temp ← temp - 4
                                        VirtualMemory[temp] ← GPR[22]
                                endif
                                temp ← temp - 4
                                VirtualMemory[temp] ← GPR[21]
                        endif
```

FIG. 8A

EXTENDED SAVE INSTRUCTION

```
        temp ← temp - 4
                        VirtualMemory[temp] ← GPR[20]
                endif
                temp ← temp - 4
                VirtualMemory[temp] ← GPR[19]
        endif
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[18]
endif
if sl = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[17]
endif
if s0 = 1 then
        temp ← temp - 4
        VirtualMemory[temp] ← GPR[16]
endif
case aregs of
        2#0000 2#0100 2#1000 2#1100 2#1110: astatic ← 0
        2#0001 2#0101 2#1001 2#1101: astatic ← 1
        2#0010 2#0110 2#1010: astatic ← 2
        2#0011 2#0111: astatic ← 3
        2#1011: astatic ← 4
        otherwise: UNPREDICTABLE
endcase
if astatic > 0 then
        temp ← temp - 4
        VirtualMemory [temp] ← GPR[7]
        if astatic > 1 then
                temp ← temp - 4
                VirtualMemory [temp] ← GPR[6]
                if astatic > 2 then
                        temp ← temp - 4
                        VirtualMemory [temp] ← GPR[5]
                        if astatic > 3 then
                                temp ← temp - 4
                                VirtualMemory [temp] ← GPR[4]
                        endif
                endif
        endif
endif
temp ← temp2 - (0 || (framesize << 3 ))
GPR[29] ← temp
```

FIG. 8B

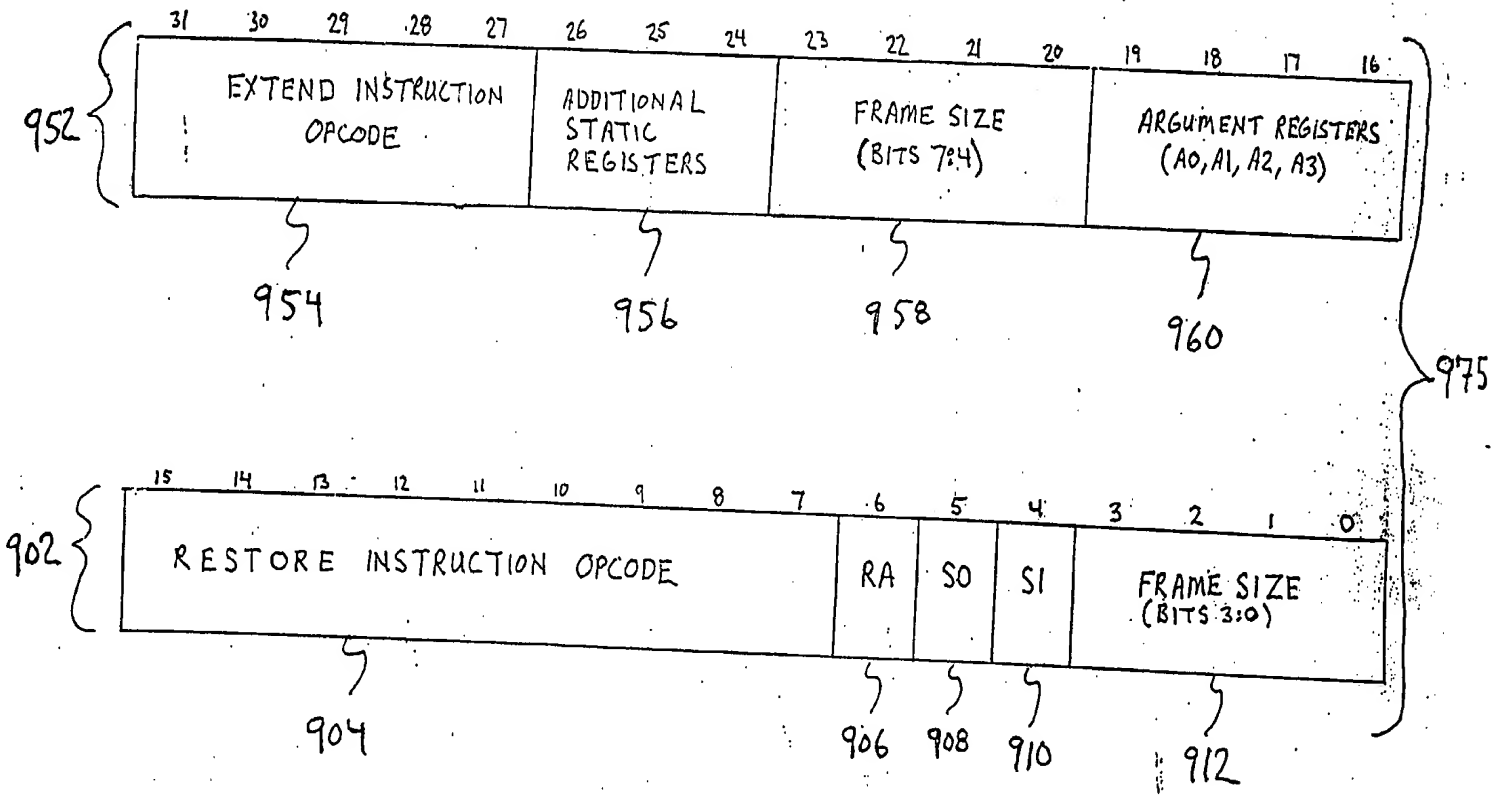
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
952
EXTEND INSTRUCTION OPCODE
ADDITIONAL STATIC REGISTERS
FRAME SIZE (BITS 7:4)
ARGUMENT REGISTERS (A0, A1, A2, A3)
954
956
958
960
975
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
902
RESTORE INSTRUCTION OPCODE
RA
S0
S1
FRAME SIZE (BITS 3:0)
904
906
908
910
912

FIG. 9

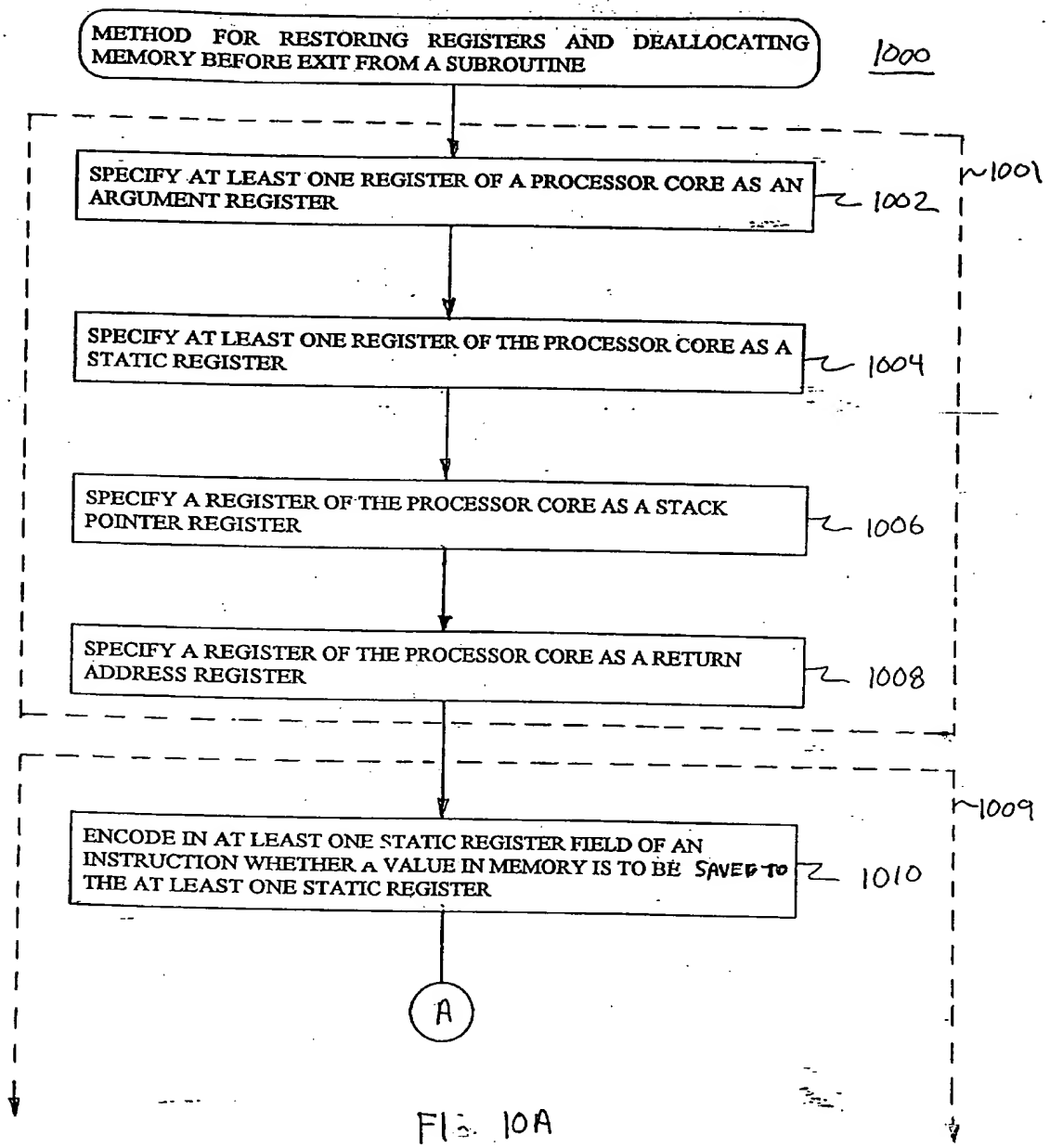
METHOD FOR RESTORING REGISTERS AND DEALLOCATING MEMORY BEFORE EXIT FROM A SUBROUTINE
1000
SPECIFY AT LEAST ONE REGISTER OF A PROCESSOR CORE AS AN ARGUMENT REGISTER
1002
1001
SPECIFY AT LEAST ONE REGISTER OF THE PROCESSOR CORE AS A STATIC REGISTER
1004
SPECIFY A REGISTER OF THE PROCESSOR CORE AS A STACK POINTER REGISTER
1006
SPECIFY A REGISTER OF THE PROCESSOR CORE AS A RETURN ADDRESS REGISTER
1008
1009
ENCODE IN AT LEAST ONE STATIC REGISTER FIELD OF AN INSTRUCTION WHETHER A VALUE IN MEMORY IS TO BE SAVED TO THE AT LEAST ONE STATIC REGISTER
1010
A

FIG. 10A

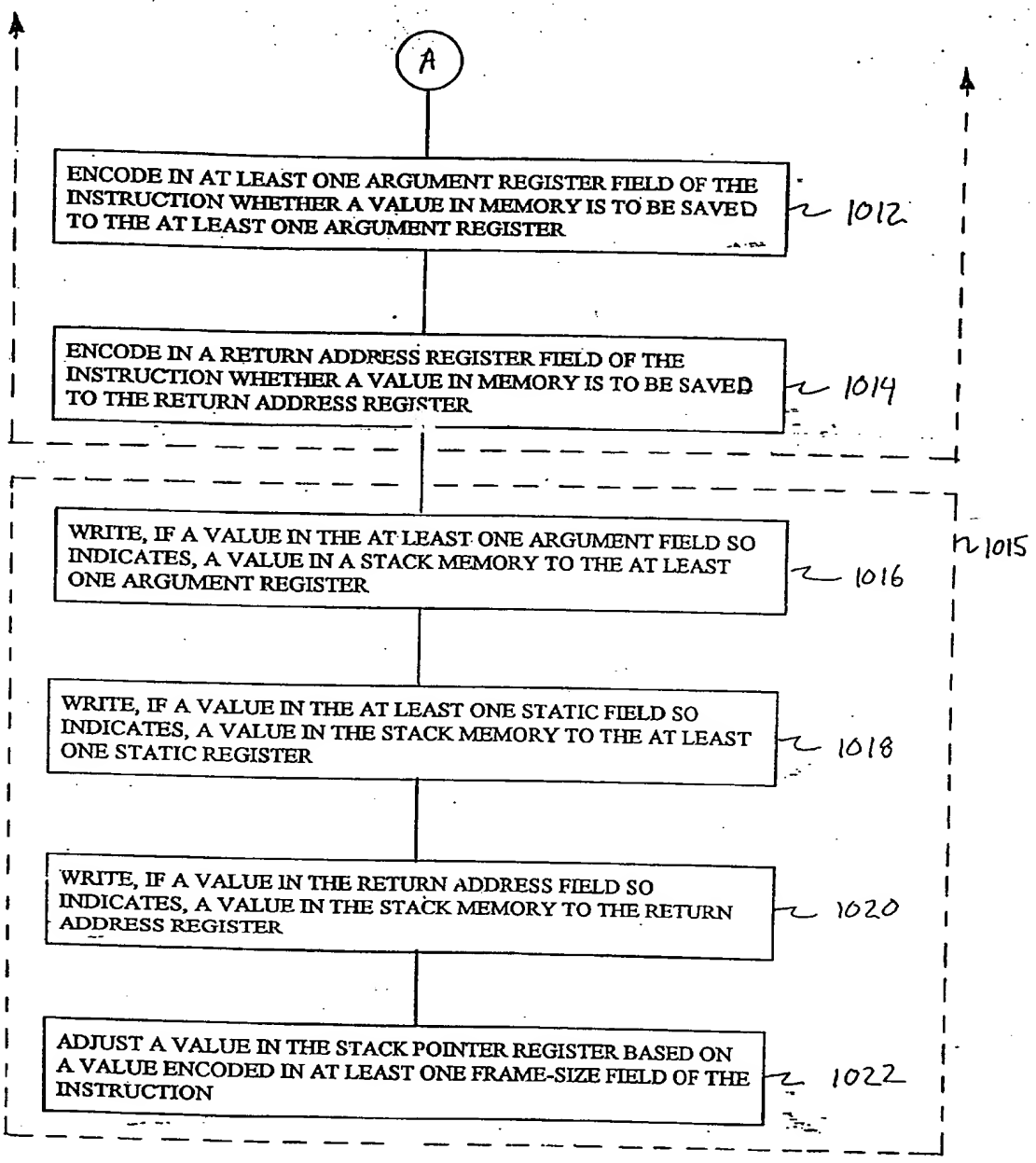
A
ENCODE IN AT LEAST ONE ARGUMENT REGISTER FIELD OF THE INSTRUCTION WHETHER A VALUE IN MEMORY IS TO BE SAVED TO THE AT LEAST ONE ARGUMENT REGISTER
1012
ENCODE IN A RETURN ADDRESS REGISTER FIELD OF THE INSTRUCTION WHETHER A VALUE IN MEMORY IS TO BE SAVED TO THE RETURN ADDRESS REGISTER
1014
1015
WRITE, IF A VALUE IN THE AT LEAST ONE ARGUMENT FIELD SO INDICATES, A VALUE IN A STACK MEMORY TO THE AT LEAST ONE ARGUMENT REGISTER
1016
WRITE, IF A VALUE IN THE AT LEAST ONE STATIC FIELD SO INDICATES, A VALUE IN THE STACK MEMORY TO THE AT LEAST ONE STATIC REGISTER
1018
WRITE, IF A VALUE IN THE RETURN ADDRESS FIELD SO INDICATES, A VALUE IN THE STACK MEMORY TO THE RETURN ADDRESS REGISTER
1020
ADJUST A VALUE IN THE STACK POINTER REGISTER BASED ON A VALUE ENCODED IN AT LEAST ONE FRAME-SIZE FIELD OF THE INSTRUCTION
1022

FIG. 10B

1100

RESTORE INSTRUCTION

```
if framesize = 0 then
        temp ← GPR[29] + 128
else
        temp ← GPR[29] + (0 || (framesize << 3 ))
endif
temp2 ← temp
if ra = 1 then
        temp ← temp - 4
        GPR [31] ← VirtualMemory [temp]
endif
if sl = 1 then
        temp ← temp - 4
        GPR [17] ← VirtualMemory [temp]
endif
if s0 = 1 then
        temp ← temp - 4
        GPR [16] ← VirtualMemory [temp]
endif
GPR[29] ← temp2
```

FIG. 11

1200

EXTENDED RESTORE INSTRUCTION

```
temp ← GPR[29] + (0 || (framesize << 3 ))
temp2 ← temp
if ra = 1 then
        temp ← temp - 4
        GPR [31] ← VirtualMemory [temp]
endif
if xsregs > 0 then
        if xsregs > 1 then
                if xsregs > 2 then
                        if xsregs > 3 then
                                if xsregs > 4 then
                                        if xsregs > 5 then
                                                if xsregs > 6 then
                                                        temp ← temp - 4
                                                        GPR[30] ← VirtualMemory [temp]
                                                endif
                                                temp ← temp - 4
                                                GPR[23] ← VirtualMemory [temp]
                                        endif
                                        temp ← temp - 4
                                        GPR[22] ← VirtualMemory [temp]
                                endif
                                temp ← temp - 4
                                GPR[21] ← VirtualMemory [temp]
                        endif
                        temp ← temp - 4
                        GPR[20] ← VirtualMemory [temp]
                endif
                temp ← temp - 4
                GPR[19] ← VirtualMemory [temp]
        endif
        temp ← temp - 4
        GPR[18] ← VirtualMemory [temp]
endif
```

FIG. 12A

EXTENDED RESTORE INSTRUCTION

```
if sl = 1 then
        temp ← temp - 4
        GPR[17] ← VirtualMemory [temp]
endif
if s0 = 1 then
        temp ← temp - 4
        GPR[16] ← VirtualMemory [temp]
endif
case aregs of
        2#0000 2#0100 2#1000 2#1100 2#1110: astatic ← 0
        2#0001 2#0101 2#1001 2#1101: astatic ← 1
        2#0010 2#0110 2#1010: astatic ← 2
        2#0011 2#0111: astatic ← 3
        2#1011: astatic ← 4
        otherwise: UNPREDICTABLE
endcase
if astatic > 0 then
        temp ← temp - 4
        GPR[7] ← VirtualMemory [temp]
        if astatic > 1 then
                temp ← temp - 4
                GPR[6] ← VirtualMemory [temp]
                if astatic > 2 then
                        temp ← temp - 4
                        GPR[5] ← VirtualMemory [temp]
                        if astatic > 3 then
                                temp ← temp - 4
                                GPR[4] ← VirtualMemory [temp]
                        endif
                endif
        endif
endif
GPR[29] ← temp2
```

FIG. 12B

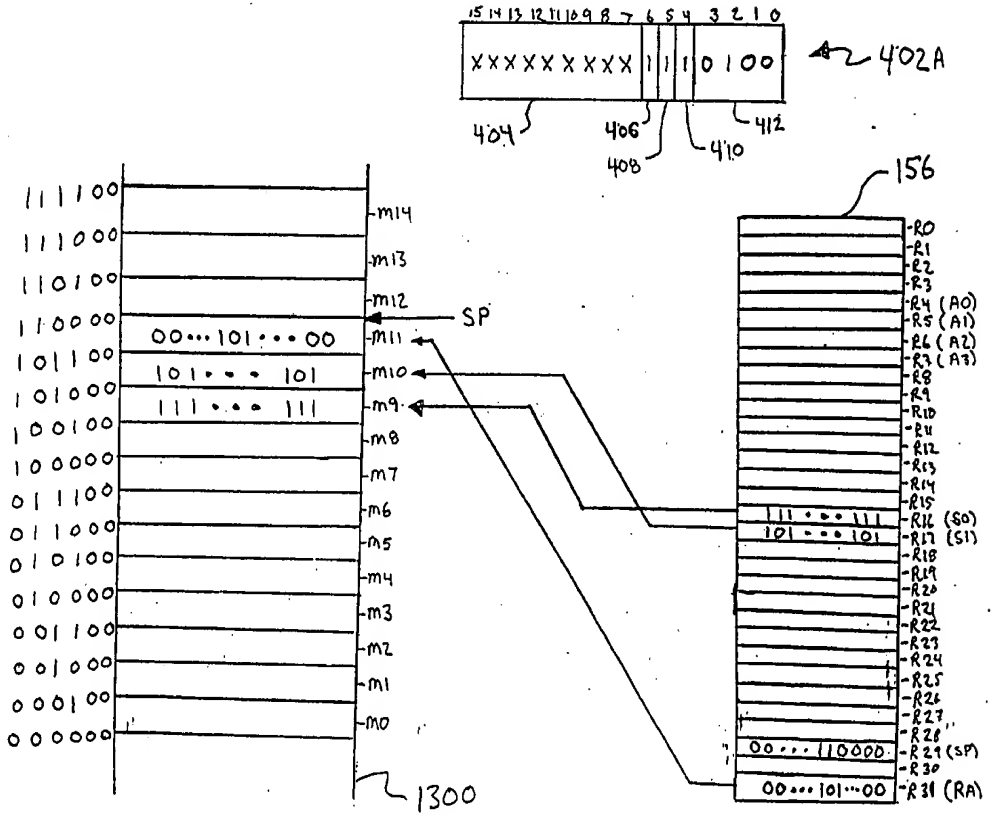
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XXXXXXXXXX 1 1 1 0 1 0 0
402A
404
406
408
410
412
156
111100
111000
110100
110000
101100
101000
100100
100000
011100
011000
010100
010000
001100
001000
000100
000000
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
SP
00…101…00
101…01
111…111
1300
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R8
R9
R10
R11
R12
R13
R14
R15
R16 (S0)
R17 (S1)
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29 (SP)
R30
R31 (RA)
111…111
101…101
00…110000
00…101…00

FIG. 13

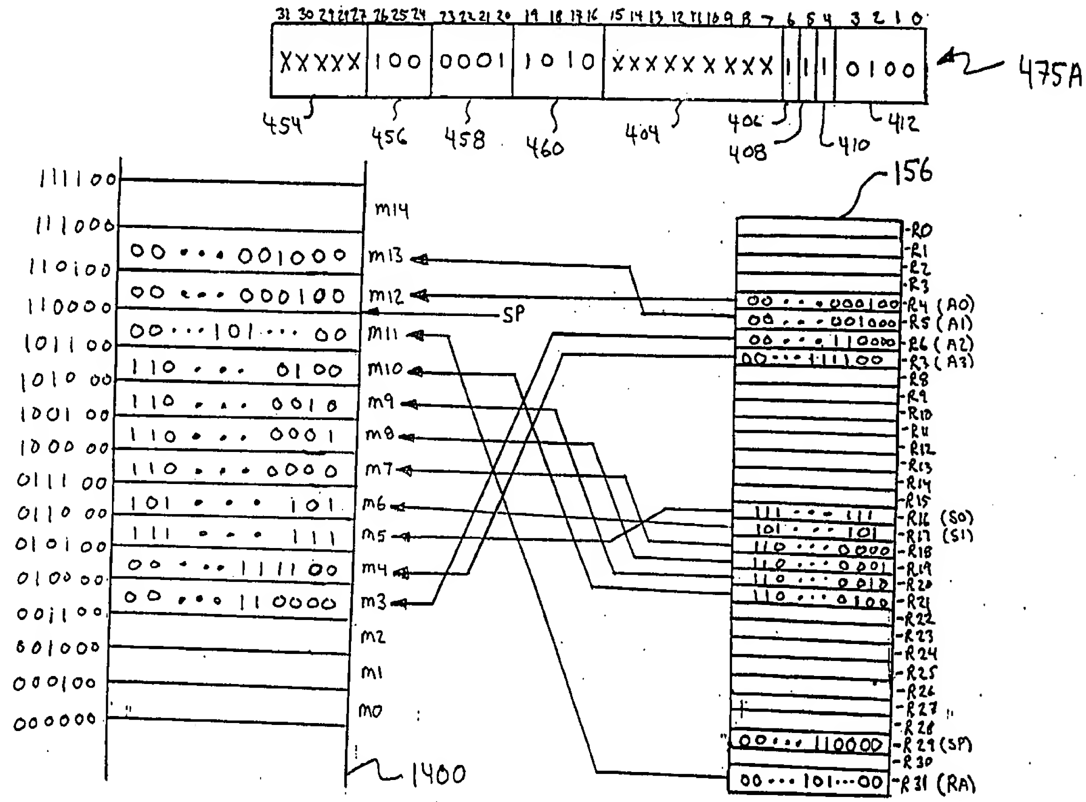
475A
454
456
458
460
404
406
408
410
412
156
1400
SP
XXXXX 100 0001 1010 XXXXXXXXX 1 1 1 0100
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R29 (SP)
R31 (RA)

FIG. 14

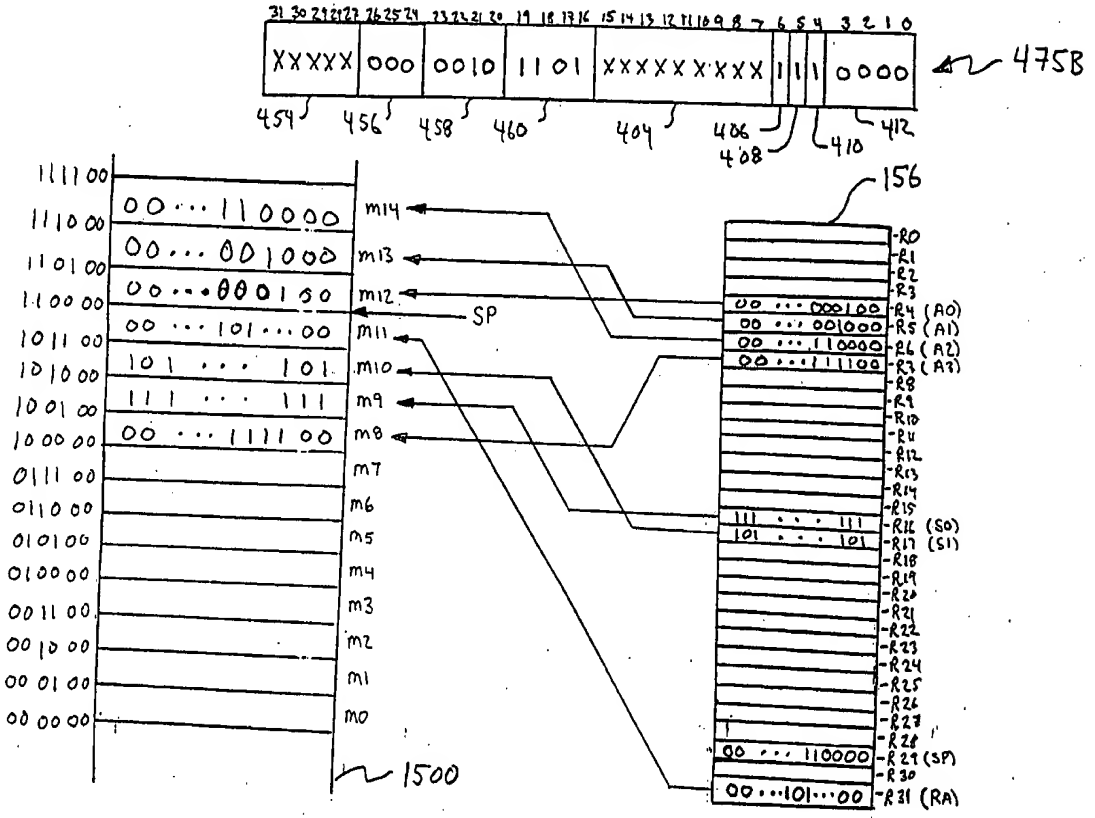
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XXXXX
000
0010
1101
XXXXXXXXXX
1 1 1
0000
475B
454
456
458
460
404
406
408
410
412
111100
111000
110100
110000
101100
101000
100100
100000
011100
011000
010100
010000
001100
001000
000100
000000
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
SP
1500
156
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R8
R9
R10
R11
R12
R13
R14
R15
R16 (S0)
R17 (S1)
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29 (SP)
R30
R31 (RA)

FIG. 15

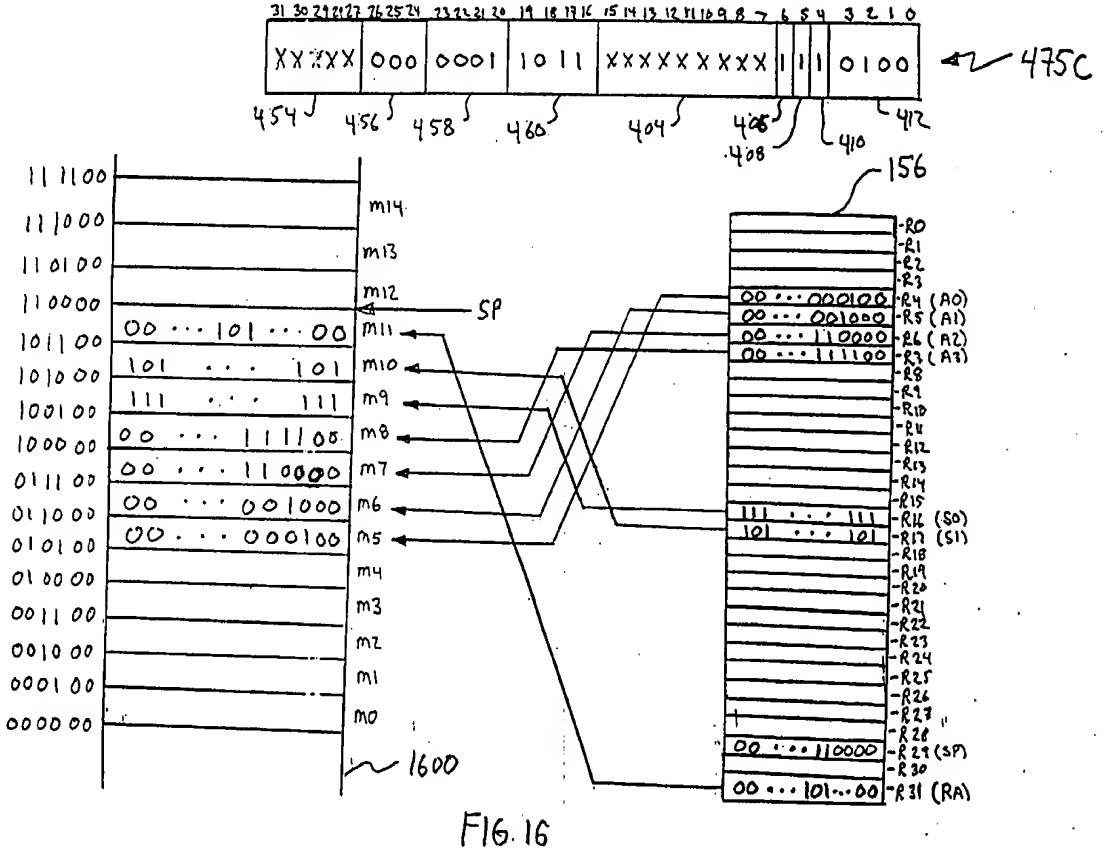
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XXXXX
000
0001
1011
XXXXXXXXXX
1 1 1 1
0100
475C
454
456
458
460
404
408
410
412
156
111100
111000
110100
110000
101100
101000
100100
100000
011100
011000
010100
010000
001100
001000
000100
000000
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
SP
00 ··· 101 ··· 00
101 ··· 101
111 ··· 111
00 ··· 111100
00 ··· 110000
00 ··· 001000
00 ··· 000100
1600
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R8
R9
R10
R11
R12
R13
R14
R15
R16 (S0)
R17 (S1)
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29 (SP)
R30
R31 (RA)
00 ··· 000100
00 ··· 001000
00 ··· 110000
00 ··· 111100
111 ··· 111
101 ··· 101
00 ··· 110000
00 ··· 101 ··· 00

FIG. 16

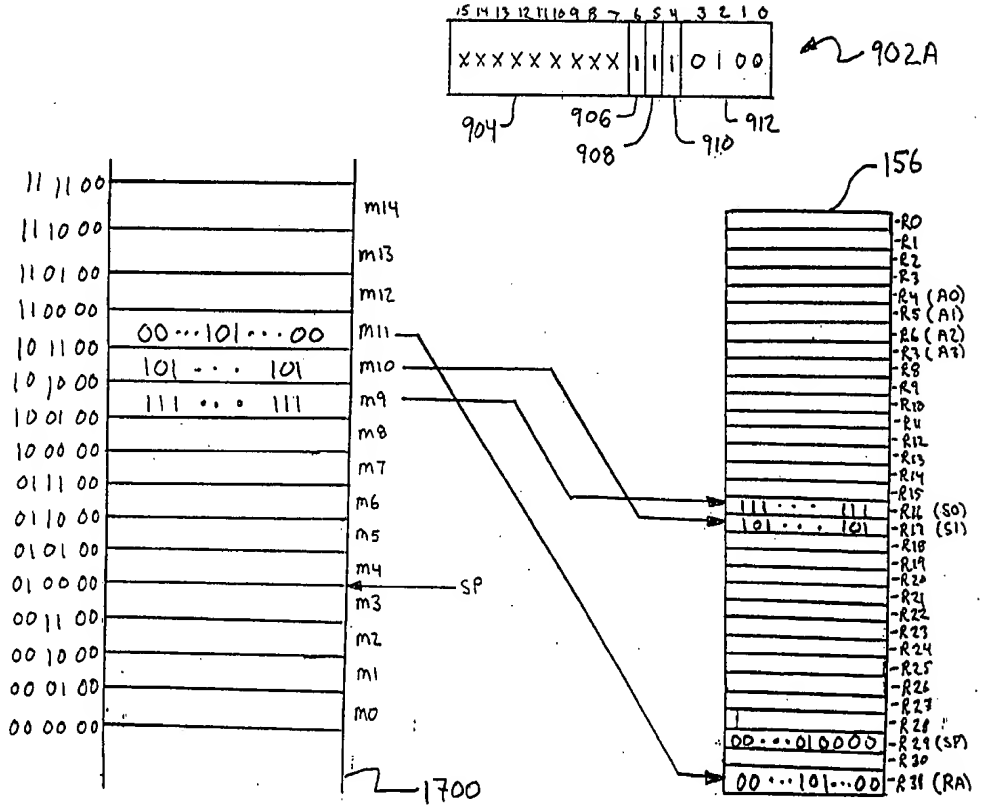
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
X X X X X X X X X 1 1 1 1 0 1 0 0
902A
904
906
908
910
912
156
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
SP
1700
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R8
R9
R10
R11
R12
R13
R14
R15
R16 (S0)
R17 (S1)
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29 (SP)
R30
R31 (RA)

FIG. 17

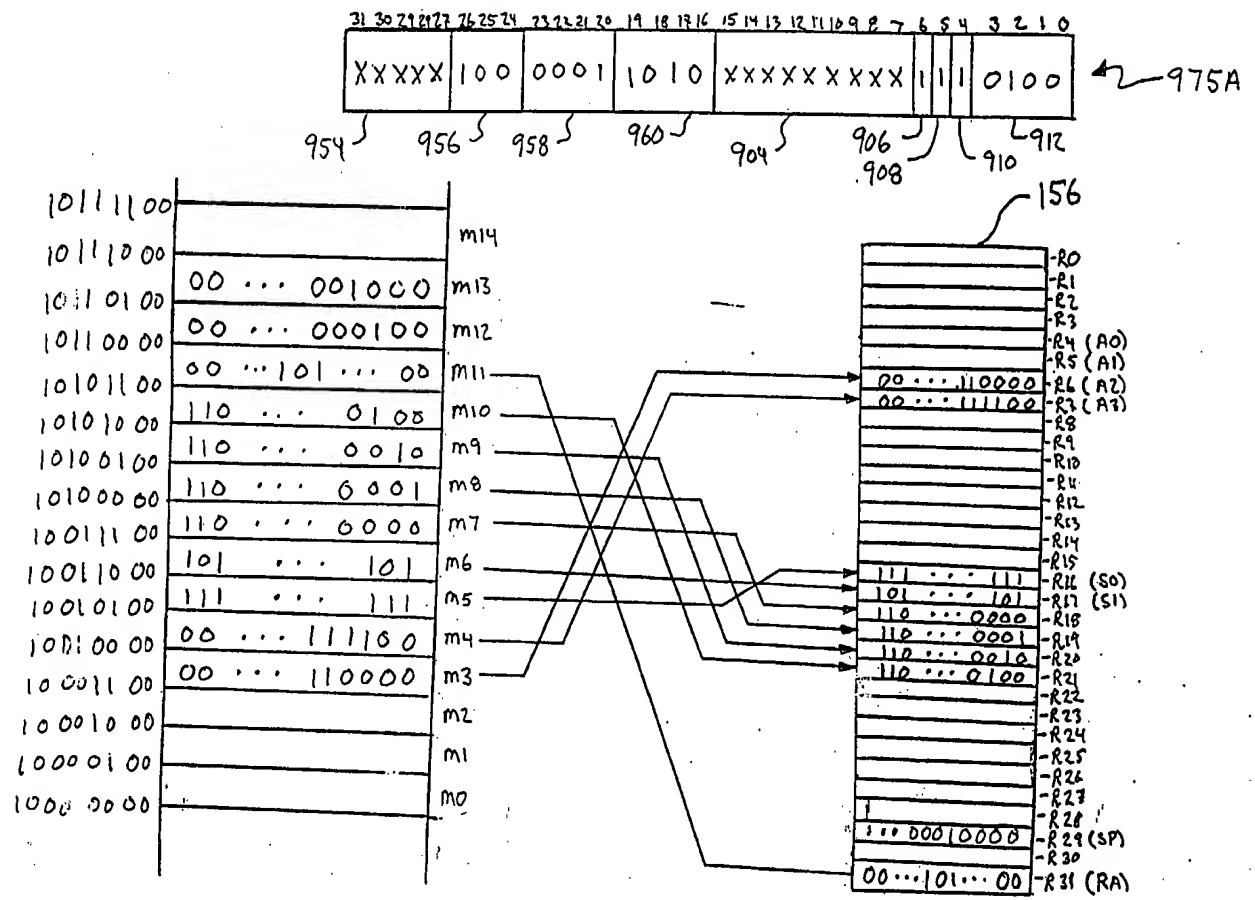
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XXXXX 100 0001 1010 XXXXXXXXXX 1 1 1 0100
975A
954
956
958
960
904
906
908
910
912
156
m14
m13
m12
m11
m10
m9
m8
m7
m6
m5
m4
m3
m2
m1
m0
R0
R1
R2
R3
R4 (A0)
R5 (A1)
R6 (A2)
R7 (A3)
R8
R9
R10
R11
R12
R13
R14
R15
R16 (S0)
R17 (S1)
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29 (SP)
R30
R31 (RA)

FIG. 18